Hierarchical Spectral Clustering

Matthew O'Malley-Nichols

Advised by Patrick Donnelly

Oregon State University Cascades

Abstract

Segmenting music based on timbral similarities provides a foundation for automated event population in music. This project addresses this need by developing a hierarchical clustering algorithm that divides songs into alternating approximate-length segments, considering both timbral and temporal similarity of individual beats. By aligning segment boundaries with song timbre, the algorithm facilitates the synchronization of events with song energy.

Feature Processing

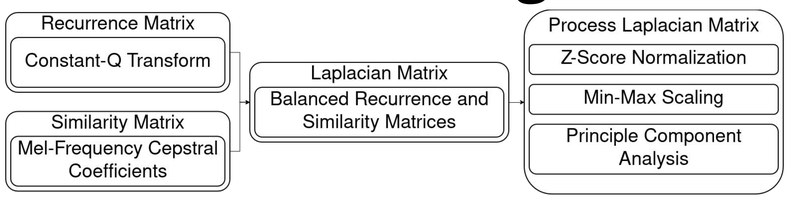

Figure 1: Feature Processing Pipeline

Before dividing the song into segments, we build recurrence and similarity matrices derived from beat frequency distributions. We combine the matrices to form a Laplacian matrix capturing the connectivity of beats. We standardize the columns of the Laplacian matrix (mean=0, SD=1) and constrain values between zero and one. Finally, we apply principal component analysis to reduce dimensionality, speeding up the runtime of our clustering algorithms.

Algorithms

Our **hierarchical clustering algorithm** (HC) aggregates song segments by minimizing a variance and length combination metric. This metric utilizes a specified variance change objective function which weights against dissimilar and oversized clusters. At each step, HC selectively merges the top nth percentile of considered clusters. The algorithm halts when clustering results in excessive variance increase or when clusters exceed the specified length threshold.

For a baseline comparison, we implement K-means along with a naive merge algorithm. The K-means algorithm optimizes timbral similarity among segments, while the merge algorithm creates segments of the desired length.

soundbendor lab

Experiments

We quantify the effectiveness of our algorithm in segmenting songs using normalized optimization metrics covering segment-length error, inter-cluster dissimilarity, intra-cluster similarity, cluster variance spread, and computational efficiency. Lower metric values indicate better performance or greater cluster validity. Experiments one and two test (n = 27) songs of Rock, Electronic, and Hip-Hop genres, while experiment three tests (n = 50) songs of mixed genres.

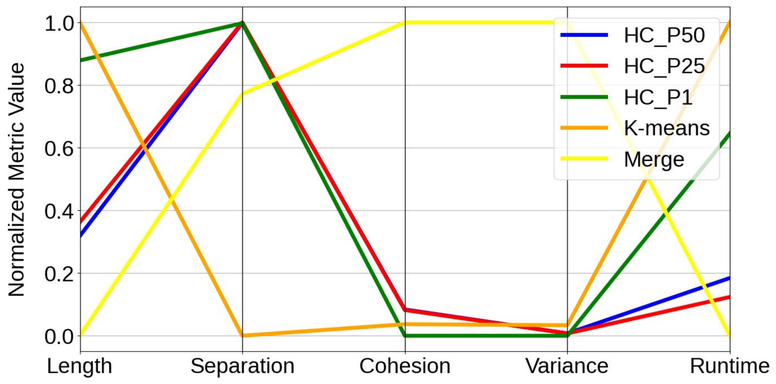

Figure 2: Experiment 1: Batch Size Test

Experiment 1 tests percentile ranges (P = 1, 25, 50) determining cluster batch size. The HC algorithm finds cohesive and low variance clusters regardless of batch size, though runtime increases with fine grained clustering.

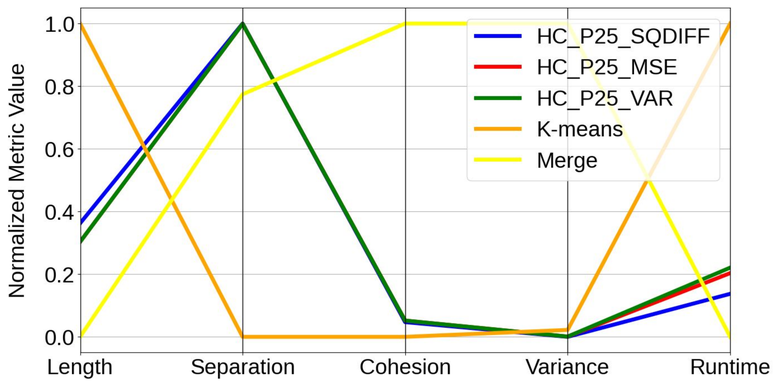

Figure 3: Experiment 2: Objective Function Evaluation

Experiment 2 compares HC objective functions: squared difference (SQDIFF), mean squared error (MSE), and squared difference (VAR) of variance change. We found no significant variance in cluster validity; however, SQDIFF exhibited a shorter runtime. This underscores the need to balance efficiency and effectiveness of objective functions.

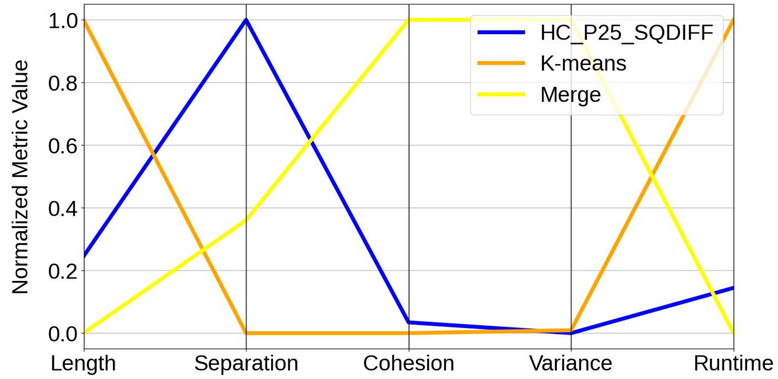

Figure 4: Experiment 3: Mixed Song Segmentation Performance

Experiment 3 evaluates optimal hyperparameters against a variety of songs. HC finds musically coherent segments similar to K-means, with lower segment-length error. The performance of HC suggests a balance of segment length and cohesion between the K-means and merge algorithms.

Discussion

The experiment results demonstrate HC's ability to obtain a balance of cluster-length accuracy and cluster cohesion while minimizing runtime, surpassing K-means in this crucial aspect. The 25th percentile clustering strategy proves the most optimal, striking a balance between cluster validity and runtime efficiency. While K-means excels in cluster separation, it compromises on cluster-length accuracy. Conversely, the merge algorithm excels in cluster-length accuracy at the expense of cluster validity. The results of experiment 3 indicate the robustness of HC to genre variations. Therefore, our HC algorithm offers a reliable approach for segmenting songs based on energy while balancing segment-length accuracy and musical cohesion.

Future Work

Future efforts using hierarchical clustering should consider alternative feature processing methods due to runtime constraints from feature extraction. Exploring alternatives could lead to pipelines capable of segmenting large datasets without impacting cluster-length accuracy or cohesion.

References

McFee, Brian, et al.. "Analyzing Song Structure with Spectral Clustering." *International Society for Music Information Retrieval Conference,* 2014

McFee, Brian, et al. "Librosa: Audio and Music Signal Analysis in python." *Proceedings of the 14th Python in Science Conference*, 2015